JUL 0 9 2004

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re application of: **Snow**	§	Group Art Unit: **2122**
	§	
Serial No.: **09/552,861**	§	Examiner: **Rutten, James D.**
	§	
Filed: **April 20, 2000**	§	Attorney Docket No.: **AUS000181US1**
	§	

For: **Application Development Server
and a Mechanism for Providing
Different Views Into the Same
Constructs Within a Strongly
Encapsulated Environment**

Certificate of Mailing Under 37 C.F.R. § 1.8(a)
I hereby certify this correspondence is being deposited with the United
States Postal Service as First Class mail in an envelope addressed to:
Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450, on
July 6, 2004.

By: _Amelia C. Turner_
Amelia C. Turner

TRANSMITTAL DOCUMENT

Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Sir:
ENCLOSED HEREWITH:

- Appellant's Brief (in triplicate) (37 C.F.R. 1.192); and
- Our return postcard.

A fee of $330.00 is required for filing an Appellant's Brief. Please charge this fee to IBM Corporation Deposit Account No. 09-0447. No additional fees are believed to be necessary. If, however, any additional fees are required, I authorize the Commissioner to charge these fees which may be required to IBM Corporation Deposit Account No. 09-0447. No extension of time is believed to be necessary. If, however, an extension of time is required, the extension is requested, and I authorize the Commissioner to charge any fees for this extension to IBM Corporation Deposit Account No. 09-0447.

Respectfully submitted,

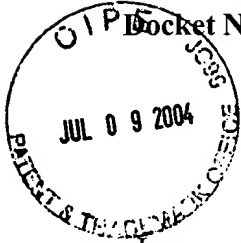_Duke W. Yee_
Duke W. Yee
Registration No. 34,285
YEE & ASSOCIATES, P.C.
P.O. Box 802333
Dallas, Texas 75380
(972) 367-2001
ATTORNEY FOR APPLICANT

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re application of: **Snow** §	
§	Group Art Unit: **2122**
Serial No.: **09/552,861** §	
§	Examiner: **Rutten, James D.**
Filed: **April 20, 2000** §	
§	
For: **Application Development Server** §	
and a Mechanism for Providing §	
Different Views Into the Same	
Constructs Within a Strongly	
Encapsulated Environment	

Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

ATTENTION: Board of Patent Appeals and Interferences

07/14/2004 HALI11 00000061 090447 09552861
01 FC:1402   330.00 DA

APPELLANT'S BRIEF (37 C.F.R. 1.192)

This brief is in furtherance of the Notice of Appeal, filed in this case on May 4, 2004.

The fees required under § 1.17(c), and any required petition for extension of time for filing this brief and fees therefore, are dealt with in the accompanying TRANSMITTAL OF APPEAL BRIEF.

This brief is transmitted in triplicate. (37 C.F.R. 1.192(a))

REAL PARTIES IN INTEREST

As reflected in the Assignment recorded on April 20, 2000, at Reel 010762, Frame 0015, the present application is assigned to International Business Machines Corporation, the real party in interest.

RELATED APPEALS AND INTERFERENCES

With respect to other appeals or interferences that will directly affect, or be directly affected by, or have a bearing on the Board's decision in the pending appeal, there are no such appeals or interferences.

STATUS OF CLAIMS

Claims 1, 2, 5, 6, 9, 11-13, and 25-35 stand finally rejected by the Examiner as noted in the final rejection mailed February 5, 2004.

STATUS OF AMENDMENTS

Applicant's Supplemental Response to Office Action, transmitted on April 27, 2004, has been entered.

SUMMARY OF INVENTION

Applicant's claims describe a method, server, and product for providing automated software development to a client. A storage unit is described for storing a plurality of working definitions for a plurality of computing environments. An interface is described for receiving requests for one of the computing environments from a client and transferring the requested computing environment to the client over a communications link.

Each one of the computing environments includes a set of first working definitions that defines a plurality of applications and a second working definition that defines the computing environment. The set of first working definitions and the second working definition are included in the plurality of working definitions.

The second working definition defines the requested computing environment. Further, the second working definition is included in the computing environment itself.

Each one of the first working definitions includes platform independent definitions that define characteristics, of one of the plurality of applications, that are necessary to construct a valid runtime image of the application. The characteristics include state, settings, and structures that are required to build the runtime image of the application. The second working definition includes platform independent definitions that define characteristics of the computing environment that are necessary to construct a valid runtime image of the computing environment. The characteristics include state, settings, and structures required to build the runtime image of the computing environment.

According to dependent claims, each one of the first working definitions includes source code for one of the applications defined by the working definition.

According to dependent claims, encrypted links are sent to the client along with the requested computing environment where the encrypted links provide support for development, testing, beta testing, and deployment of the plurality of applications.

According to dependent claims, the interface receives an update to one of the first working definitions. That working definition is then modified to reflect the update.

According to dependent claims, a change occurs in the data processing system. In response to a determination that the change affects one of the working definitions, that working definition is modified to reflect the change. The runtime image of the application defined by that working definition is updated.

According to dependent claims, the change is an event.

According to dependent claims, the communications link is the Internet.

According to dependent claims, the communications link is an intranet.

ISSUES

Are the Examiner's rejections of claims 1, 2, 5, 9, 25, 26, 30, and 31 under 35 U.S.C. § 102(e) as being anticipated by U.S. Patent 6,145,119 issued to *House*; claims 6, 27, and 32 under 35 U.S.C. § 103(a) as being unpatentable over *House* in view of U.S. Patent 5,423,042 issued to *Jalili*; claims 1, 2, 11-13, 25, 28-30, and 33-35 under 35 U.S.C. § 103(a) as being unpatentable over U.S. Patent 5,361,360 issued to *Ishigami* in view of U.S. Patent 6,083,276 issued to *Davidson*; claims 5, 9, 26, and 31 under 35 U.S.C. § 103(a) as being unpatentable over *Ishigami* and *Davidson* in view of U.S. Patent 5,953,525 issued to *Glaser*; and claims 6, 27, and 32 under 35 U.S.C. § 103(a) as being unpatentable over *Ishigami* and *Davidson* in view of *Jalili* well founded?

GROUPING OF CLAIMS

For the purposes of this appeal, claims 1, 2, 5, 6, 9, 11-13, and 25-35 stand or fall together as one group.

ARGUMENT

The Examiner rejected claims 1, 2, 5, 9, 25, 26, 30, and 31 under 35 U.S.C. § 102(e) as being anticipated by U.S. Patent 6,145,119 issued to *House*. This position is not well founded.

According to Applicant's claims, each computing environment includes a set of first working definitions that define a plurality of applications. Each computing environment also includes a second working definition that defines the computing environment.

Each first working definition includes platform independent definitions that define the characteristics of one of the applications. These are characteristics that are necessary to construct a valid runtime image of the application, including state, settings, and structures that are required in order to build the runtime image.

The second working definition includes platform independent definitions that define the characteristics of its computing environment. These are characteristics that are necessary to

construct a valid runtime image of the computing environment, including state, settings, and structures that are required in order to build the runtime image. The second working definition is included in the computing environment itself.

House does not teach (1) a plurality of computing environments, (2) a computing environment that includes a set of first working definitions that define a plurality of applications, (3) a computing environment that includes a second working definition that defines the computing environment itself, (4) a second working definition that defines characteristics that are necessary to construct a valid runtime image of its computing environment, or (5) receiving a request for a computing environment where the computing environment includes a set of first working definitions that define a plurality of applications and a second working definition that defines the computing environment itself.

House describes providing a programming environment that supports development of internet applications. *House* describes a data structure for storing all aspects of a project. The data structure is for encapsulating a software project with a project application that is defined by executable programming logic, and a project environment for developing the application. Each data structure therefore defines one software project. Each data structure includes one application that includes executable logic. Each data structure also includes descriptive data that describes the project environment.

House does not teach a plurality of computing environments. *House* teaches projects. The Examiner states that a project is a computing environment. A project, however, is not a computing environment. An "environment" is defined by Microsoft Press Computer Dictionary, Third Edition, published 1997, as "the configuration of resources available to the user. Environment refers to the hardware and the operating system running on it." Applicant describes a computing environment. Applicant also describes constructing a runtime image of that computing environment. A "project" is not a "computing environment". Therefore, *House* does not teach a plurality of computing environments.

House does not teach a computing environment that includes a set of first working definitions that define a plurality of applications. Applicant claims a computing environment that includes a set of first working definitions that define a plurality of applications. According to *House*, a first section of the structure includes the executable program logic while a second section of the structure includes all other data that is needed. An APP file is constructed that

includes all of the data and structures that represent a software project including project related data such as information on the application for the server, stored procedures or user defined functions, source code, event logic controls, forms, and relationships among all of the things. From the single APP file 454, a project and all of its state can be recovered.

The APP file 454 includes a runtime section 550 and an "other data" section 552. The runtime section contains all of the data that is needed at runtime to run an application. The "other data" section includes all other project related data. When executing an application, the "other data" section can be skipped.

The Examiner states that *House* teaches the set of first working definitions by teaching the runtime section 550 of APP file 454, and teaches a second working definition by teaching the "other data" section 552 of APP file 454.

House teaches a data structure which includes two parts, a runtime section and an "other data" section. The runtime section can be used to run one application. *House* does teach multiple data structures. However, each data structure is for a different project. If, as the Examiner argues, a project is analogous to a computing environment, one project must include multiple data structures in order to teach a computing environment that includes a set of first working definitions. On the contrary, *House* teaches a single data structure being used to encapsulate all elements that are necessary to build one project. Therefore, there is one data structure to represent one project.

House does not teach a computing environment that includes a second working definition that defines the computing environment itself. Following the Examiner's argument that "project" teaches a computing environment, the "project" itself must include a second working definition that defines the "project". Further, following the Examiner's argument, because the "other data" section is believed by the Examiner to be the second working definition, the "other data" section must be included in the "project". The "other data" section, however, is not included in the project. The "other data" section includes data required to restore the project environment. Applicant claims a computing environment including a working definition that defines that computing environment. There is no "project environment" in *House* that includes this "other data" section. The "other data" section is used to store data that is needed to restore the project environment. The project environment does not exist apart from the "other data" section.

House does not teach a second working definition that defines characteristics that are necessary to construct a valid runtime image of its computing environment. According to Applicant's claims, both the first working definitions and the second working definition include definitions that are necessary in order to construct a valid runtime image. *House* does not teach a set of first working definitions and a second working definition that all include definitions that are necessary to construct a valid runtime image.

Data objects that comprise executable programming logic are stored in the runtime section 550. The remaining data that describes a development environment is stored in the "other data" section 552. In this manner, the runtime data is kept separate (kept in runtime section 550) from the descriptive, non-runtime, data (kept in "other data" section 552). When executing an application, reading in the "other data" section 552 is skipped. Therefore, it is clear that the "other data" section does not include runtime data.

House teaches away from the "other data" section including definitions that are necessary to construct a runtime image. *House* teaches the runtime data being included in the runtime section 550 and not in the "other data" section 552 that the Examiner states teaches Applicant's "second working definition".

House does not teach receiving a request for a computing environment. *House* does not teach receiving a request for a computing environment where the computing environment includes a set of first working definitions that define a plurality of applications and a second working definition that defines the computing environment itself.

The Examiner rejected claims 6, 27, and 32 under 35 U.S.C. § 103(a) as being unpatentable over *House* in view of U.S. Patent 5,423,042 issued to *Jalili*. This position is not well founded.

These claims describe encrypted links that are sent to the client along with the requested computing environment where the encrypted links provide support for development, testing, beta testing, and deployment of the plurality of applications.

Jalili teaches remote procedure execution. The Examiner states that *Jalili* teaches a system of communication using encrypted links. Although *Jalili* does teach an encrypted identifier, *Jalili* does not supply all of the missing features of Applicant's claims. Applicant's claims do not merely describe encrypted links. These claims describe the encrypted links providing support for development, testing, beta testing, and deployment of the plurality of applications. These applications are defined by the set of first working definitions included in the requested

computing environment.

The references, neither singly nor in combination, describe, teach or suggest encrypted links that provide support for development, testing, beta testing, and deployment of these plurality of applications. Therefore, this combination of references does not render Applicant's claims unpatentable.

The Examiner rejected claims 1, 2, 11-13, 25, 28-30, and 33-35 under 35 U.S.C. § 103(a) as being unpatentable over U.S. Patent 5,361,360 issued to *Ishigami* in view of U.S. Patent 6,083,276 issued to *Davidson*. This position is not well founded.

Ishigami teaches storing definition information of a programming environment including a set of names of tools and names of data that are necessary for developing a software. According to a first aspect of *Ishigami*, tools and data are transmitted and installed. According to a second aspect of *Ishigami*, when the definition information is changed, tools and data that are already installed are updated.

The Examiner states that *Ishigami* teaches a first set of working definitions by teaching the tools and data according to the first aspect of *Ishigami*, and teaches a second working definition by teaching the updating of already stored tools and data according to the second aspect of *Ishigami*.

Ishigami does not teach requesting a computing environment that includes a first set of working definitions and a second working definition. In *Ishigami*, definitions may be received. These definitions define tools and data. Some of these tools and data may have already been stored. Thus, these definitions may include updated version of some of the stored tools and data. *Ishigami* does not teach requesting a computing environment that includes a first set of working definitions that define characteristics of a plurality of applications and a second working definition that defines characteristics of the requested computing environment. The section of the reference referred to by the Examiner describes definition information being received that includes updates of tools and data that are already stored. This definition information is not analogous to the second working definition of Applicant's claims.

Applicant describes the second working definition defining characteristics that are used to construct a runtime image of the requested computing environment. Further, the second working definition is included in the computing environment itself.

According to Applicant's claims, a computing environment is requested. This computing environment includes a set of first working definitions and a second working definition that defines characteristics of the computing environment itself. The definition information stored according to the second aspect of *Ishigami*, is not analogous to the second working definition of Applicant's claims. The stored definition information of *Ishigami* describes the tools and data already stored. These tools and data are compared to a change in the definition information that was received in response to a request. Thus, in response to a hardware configuration change, tools and data are automatically re-installed. The definition information according to the second aspect of *Ishigami* does not teach a second working definition that defines characteristics of the definition itself.

The definition information of *Ishigami* does not describe, teach, or suggest a computing environment that includes a first set of working definitions as well as a second working definition that defines the computing environment itself. *Ishigami* teaches definition information that includes names of tools and data. Information may also be stored in the requesting system relating to a configuration that is already stored. The stored information can be compared to requested definition information. Already installed tools and data that were changed can then be re-installed. Nothing in *Ishigami* teaches a requested computing environment that includes a set of first working definitions and a second working definition.

The Examiner states that *Ishigami* does not teach definitions that are necessary to build a runtime image. The Examiner uses *Davidson* to supply these missing features. *Davidson* teaches creating and configuring a component-based application through descriptive attribute grammar.

Davidson does not teach a computing environment that includes a first set of working definitions that define a plurality of applications, and a second working definition that defines the computing environment itself. As described above, *Ishigami* also does not teach a computing environment that includes a first set of working definitions that define a plurality of applications, and a second working definition that defines the computing environment itself. Neither reference describes, teaches, or suggests these features. Therefore, the combination of the references does not describe, teach, or suggest Applicant's claims.

Neither reference describes, teaches, or suggests a computing environment that includes a first set of working definitions that define a plurality of applications, and a second working

definition that defines the computing environment itself where the first set of working definitions defines characteristics that are necessary to construct a valid runtime image of the applications and the second working definition defines characteristics that are necessary to construct a valid runtime image of the computing environment. Therefore, this combination of references does not render Applicant's claims unpatentable.

The Examiner rejected claims 5, 9, 26, and 31 under 35 U.S.C. § 103(a) as being unpatentable over *Ishigami* and *Davidson* in view of U.S. Patent 5,953,525 issued to *Glaser*. This position is not well founded.

The Examiner states that the combination of *Ishigami* and *Davidson* does not describe the working definition including source code and uses *Glaser* to supply this missing feature. The Examiner also states that the combination of *Ishigami* and *Davidson* does not describe the communication line being an intranet and uses *Glaser* to supply this missing feature.

The combination of *Ishigami*, *Davidson*, and *Glaser* does not describe, teach, or suggest a computing environment that includes a first set of working definitions that define a plurality of applications, and a second working definition that defines the computing environment itself where the first set of working definitions defines characteristics that are necessary to construct a valid runtime image of the applications and the second working definition defines characteristics that are necessary to construct a valid runtime image of the computing environment, and where each one of the first working definitions include source code for the application.

The combination of *Ishigami*, *Davidson*, and *Glaser* does not describe, teach, or suggest a computing environment that includes a first set of working definitions that define a plurality of applications, and a second working definition that defines the computing environment itself where the first set of working definitions defines characteristics that are necessary to construct a valid runtime image of the applications and the second working definition defines characteristics that are necessary to construct a valid runtime image of the computing environment, and where the communications line is an intranet.

Therefore, the combination of *Ishigami*, *Davidson*, and *Glaser* does not render Applicant's claims unpatentable.

The Examiner rejected claims 6, 27, and 32 under 35 U.S.C. § 103(a) as being unpatentable over *Ishigami* and *Davidson* in view of *Jalili*. This position is not well founded.

These claims describe encrypted links that are sent to the client along with the requested computing environment where the encrypted links provide support for development, testing, beta testing, and deployment of the plurality of applications.

The Examiner states that the combination of *Ishigami* and *Davidson* does not teach sending encrypted links. As discussed above, although *Jalili* does teach an encrypted identifier, *Jalili* does not supply all of the missing features of these claims. Applicant's claims do not merely describe encrypted links. These claims describe the encrypted links providing support for development, testing, beta testing, and deployment of the plurality of applications. These applications are defined by the set of first working definitions included in the requested computing environment. Because *Jalili* does not teach encrypted links providing support for development, testing, beta testing, and deployment of the plurality of applications, and because neither *Ishigami* nor *Davidson* teach a second working definition that defines the computing environment in which the second working definition is included, where the second working definition define characteristics that are necessary to construct a runtime image, the combination of *Ishigami*, *Davidson*, and *Jalili* does not render Applicant's claims unpatentable.

Applicant's claims are patentably distinct over the cited prior art. *House* does not teach (1) a plurality of computing environments, (2) a computing environment that includes a set of first working definitions that define a plurality of applications, (3) a computing environment that includes a second working definition that defines the computing environment itself, (4) a second working definition that defines characteristics that are necessary to construct a valid runtime image of its computing environment, or (5) receiving a request for a computing environment where the computing environment includes a set of first working definitions that define a plurality of applications and a second working definition that defines the computing environment itself.

Lisa L.B. Yociss
Reg. No. 36,975
Yee & Associates, P.C.
PO Box 802333
Dallas, TX 75380
(972) 367-2001

APPENDIX OF CLAIMS

The text of the claims involved in the appeal reads:

1.   An application development server, comprising:

a storage unit for storing a plurality of working definitions for a plurality of computing environments;

an interface for receiving requests for one of the plurality of computing environments from and transferring the requested one of the plurality of computing environments to a client over a communications line;

each one of said plurality of computing environments including a set of first working definitions that define a plurality of applications and a second working definition that defines said each one of said plurality of computing environments, said set of first working definitions and said second working definition being included in said plurality of working definitions;

each one of said set of first working definitions including platform independent definitions that define characteristics of one of said plurality of applications that are necessary to construct a valid runtime image of said one of said plurality of applications, said characteristics including state, settings, and structures required to build said runtime image of said one of said plurality of applications; and

said second working definition including platform independent definitions that define characteristics of said each one of said plurality of computing environments that are necessary to construct a valid runtime image of said each one of said plurality of computing environments, said characteristics including state, settings, and structures required to build said runtime image of said each one of said plurality of computing environments.

2.   The application development server as recited in claim 1, wherein the communication

line is the Internet.

5.   The application development server as recited in claim 1, wherein each one of said set of

first working definitions includes source code for one of said plurality of applications defined by

said one of said set of first working definitions.

6.   The application development server as recited in claim 1, wherein encrypted links are sent

to the client along with the requested one of the plurality of computing environments, the

encrypted links providing support for development, testing, beta testing, and deployment of the

plurality of applications.

9.   The application development server as recited in claim 1, wherein the communications

line is an intranet.

11.   The application development server as recited in claim 1, further comprising:

said interface receiving an update to one of said set of first working definitions; and

said one of said set of first working definitions being modified to reflect said update.

12.   The application development server as recited in claim 1, further comprising:

a change occurring within a data processing system;

responsive to a determination that the change affects one of said set of first working definitions, the one of said set of first working definitions being modified to reflect the change; and

the runtime image of one of said plurality of applications defined by said one of said set of first working definitions being updated.

13.   The application development server as recited in claim 12, wherein the change is an event.

25.   A computer program product, comprising:

instruction means for storing a plurality of working definitions for a plurality of computing environments;

instruction means for receiving requests for one of the plurality of computing environments from and transferring the requested one of the plurality of computing environments to a client over a communications line;

each one of said plurality of computing environments including a set of first working definitions that define a plurality of applications and a second working definition that defines said each one of said plurality of computing environments, said set of first working definitions and said second working definition being included in said plurality of working definitions;

each one of said set of first working definitions including platform independent definitions that define characteristics of one of said plurality of applications that are necessary to construct a valid runtime image of said one of said plurality of applications, said characteristics

including state, settings, and structures required to build said runtime image of said one of said plurality of applications; and

said second working definition including platform independent definitions that define characteristics of said each one of said plurality of computing environments that are necessary to construct a valid runtime image of said each one of said plurality of computing environments, said characteristics including state, settings, and structures required to build said runtime image of said each one of said plurality of computing environments.

26. The product as recited in claim 25, wherein each one of said set of first working definitions includes source code for one of said plurality of applications defined by said one of said set of first working definitions.

27. The product as recited in claim 25, further comprising instruction means for sending encrypted links to the client along with the requested one of the plurality of computing environments, the encrypted links providing support for development, testing, beta testing, and deployment of the plurality of applications.

28. The product as recited in claim 25, further comprising:

instruction means for receiving an update to one of said set of first working definitions; and

instruction means for modifying said one of said set of first working definitions to reflect said update.

29.   The product as recited in claim 25, further comprising:

instruction means for detecting a change within a data processing system;

responsive to a determination that the change affects one of said set of first working definitions, instruction means for modifying the one of said set of first working definitions to reflect the change; and

instruction means for updating the runtime image of one of said plurality of applications defined by said one of said set of first working definitions.

30.   A method for providing automated software development to a client, said method comprising:

storing a plurality of working definitions for a plurality of computing environments;

receiving requests for one of the plurality of computing environments from and transferring the requested one of the plurality of computing environments to a client over a communications line;

each one of said plurality of computing environments including a set of first working definitions that define a plurality of applications and a second working definition that defines said each one of said plurality of computing environments, said set of first working definitions and said second working definition being included in said plurality of working definitions;

each one of said set of first working definitions including platform independent definitions that define characteristics of one of said plurality of applications that are necessary to construct a valid runtime image of said one of said plurality of applications, said characteristics including state, settings, and structures required to build said runtime image of said one of said plurality of applications; and

said second working definition including platform independent definitions that define characteristics of said each one of said plurality of computing environments that are necessary to construct a valid runtime image of said each one of said plurality of computing environments, said characteristics including state, settings, and structures required to build said runtime image of said each one of said plurality of computing environments.

31.   The method as recited in claim 31, wherein each one of said set of first working definitions includes source code for one of said plurality of applications defined by said one of said set of first working definitions.

32.   The method as recited in claim 31, further comprising sending encrypted links to the client along with the requested one of the plurality of computing environments, the encrypted links providing support for development, testing, beta testing, and deployment of the plurality of applications.

33.   The method as recited in claim 31, further comprising:
receiving an update to one of said set of first working definitions; and
modifying said one of said set of first working definitions to reflect said update.

34.   The method as recited in claim 31, further comprising:
detecting a change within a data processing system;
responsive to a determination that the change affects one of said set of first working definitions, modifying the one of said set of first working definitions to reflect the change; and

updating the runtime image of one of said plurality of applications defined by said one of said set of first working definitions.

35. The method as recited in claim 34, wherein the change is an event.